



SNS COLLEGE OF ENGINEERING



Kurumbapalayam(Po), Coimbatore - 641 107

Accredited by NAAC-UGC with 'A' Grade

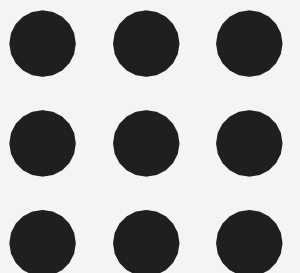
Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

Department of Information Technology

19CS204 OBJECT ORIENTED PROGRAMMING

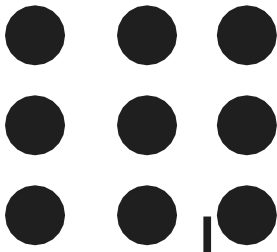
I YEAR /II SEMESTER

Topic - Creating Thread





Multithreading



- In the most general sense, you create a thread by instantiating an object of type Thread.
- Java defines two ways in which this can be accomplished:
 - implement the Runnable interface.
 - extend the Thread class, itself.



Multithreading



implement the Runnable interface.

- The easiest way to create a thread is to create a class that implements the Runnable interface.
- We create a new class which implements java.lang.Runnable interface and override run() method. Then we instantiate a Thread object and call start() method on this object.
- To implement Runnable, a class need only implement a single method called run(), which is declared like this:

```
public void run( )
```

- Inside run(), you will define the code that constitutes the new thread.



Multithreading



implement the Runnable interface.

Example

```
public class Multithread implements Runnable
{
    public void run()
    {
        System.out.println("thread is running...");
    }
    public static void main(String args[]){
        Multithread m1=new Multithread();
        Thread t1 =new Thread(m1);
        t1.start();
    }
}
```



Multithreading



implement the Runnable interface.

```
class MultithreadingDemo implements Runnable {
    public void run()
    {
        try {
            // Displaying the thread that is running
            System.out.println(
                "Thread " + Thread.currentThread().getId()
                + " is running");
        }
        catch (Exception e) {
            // Throwing an exception
            System.out.println("Exception is caught");
        }
    }
}
```

```
// Main Class
class Multithread {
    public static void main(String[] args)
    {
        int n = 8; // Number of threads
        for (int i = 0; i < n; i++) {
            Thread object
                = new Thread(new MultithreadingDemo());
            object.start();
        }
    }
}
```



Multithreading



implement the Runnable interface.

```
class NewThread implements Runnable {
    Thread t;
    NewThread() {
        // Create a new, second thread
        t = new Thread(this, "Demo Thread");
        System.out.println("Child thread: " + t);
        t.start(); // Start the thread
    }
    // This is the entry point for the second thread.
    public void run() {
        try {
            for(int i = 5; i > 0; i--) {
                System.out.println("Child Thread: " + i);
                Thread.sleep(500);
            }
        } catch (InterruptedException e) {
            System.out.println("Child interrupted.");
        }
        System.out.println("Exiting child thread.");
    }
}
```

```
class ThreadDemo {
    public static void main(String args[ ]) {
        new NewThread(); // create a new thread
        try {
            for(int i = 5; i > 0; i--) {
                System.out.println("Main Thread: " + i);
                Thread.sleep(1000);
            }
        } catch (InterruptedException e) {
            System.out.println("Main thread interrupted.");
        }
        System.out.println("Main thread exiting.");
    }
}
```



Multithreading

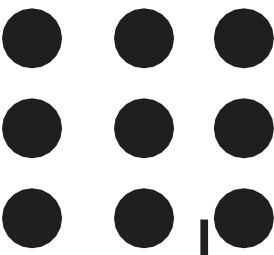


extend the Thread class

- The second way to create a thread is to create a new class that extends Thread, and then to create an instance of that class.
- We create a class that extends the java.lang.Thread class.
- The extending class must override the run() method, which is the entry point for the new thread.
- It must also call start() to begin execution of the new thread.
- Start() invokes the run() method on the Thread object.



Multithreading

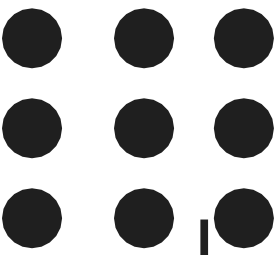


extend the Thread class

```
public class MultithreadingDemo extends Thread{  
    public void run(){  
        System.out.println("My thread is in running state.");  
    }  
  
    public static void main(String args[]){  
        MultithreadingDemo obj=new MultithreadingDemo();  
        obj.start();  
    }  
}
```




Multithreading



extend the Thread class

```
class MultithreadingDemo extends Thread {
    public void run()
    {
        try {
            // Displaying the thread that is running
            System.out.println( "Thread " + Thread.currentThread().getId()+ " is running");
        }
        catch (Exception e) {
            // Throwing an exception
            System.out.println("Exception is caught");
        } } }
// Main Class
public class Multithread {
    public static void main(String[] args)
    {
        int n = 8; // Number of threads
        for (int i = 0; i < n; i++) {
            MultithreadingDemo object = new MultithreadingDemo();
            object.start();
        } } }
```



THANK YOU